**PROJECT REPORT**

**Name: Keerthi Reddy Tummalapelly**

**Overall delay:**

Overall delay refers to the total time it takes for data to travel from the source to the destination, encompassing various components such as propagation delay, transmission delay, processing delay and queuing delay.

**Throughput:**

Throughput is calculated as the number of bits delivered per unit time.

**Overall delay and throughput calculation using the Wireshark traces for TCP protocol:**

The process I followed for calculating the overall delay of data transfer over TCP protocol involves the following steps:

1. Open the Wireshark capture file: Begin by opening the Wireshark traces for the specific file you wish to analyze.

2. Apply a filter: Filter the traces using the port number on which the server is operating to narrow down the packets to the specific communication (tcp.port == 20000)

3. Identify the initial ACK packet: Find the packet where the server initiates the data transfer to the client. This is the first packet in the sequence.

4. Identify the final ACK packet: Find the packet in which the client acknowledges the data transferred by the server.

5. Calculate overall delay: Determine the overall delay of data transfer by subtracting the timestamp of the initial packet from the timestamp of the final packet.

6. Calculate total data transferred: Sum up the payload lengths of all the data packets between initial packet and final packet, excluding the payloads of acknowledgement.

7. Calculate the throughput: Determine the throughput using the below formula

Throughput = Amount of data transferred/Total time taken

= Amount of data transferred/Overall delay

**Overall delay of File 1.txt:**

According to the Wireshark traces of File 1.txt (16 KB):

The initial packet is packet 24.

The final Packet is packet 37.

Overall Delay = Timestamp of packet 37 − Timestamp of packet 24

= 2.333360 - 2.332610

= 0.00075 = 0.0007 seconds

**Throughput of File 1. Txt:**

Total data transferred = 14,878 bytes

Overall delay = 0.0007 seconds

Throughput **=** Amount of data transferred / Overall delay

**= (**14878 \* 8) /0.0007

= 170034285.7140 bps

**Overall delay of File 2.txt:**

According to the Wireshark traces of File 2.txt (32 KB):

Initial SYN Packet is packet 32

Final ACK Packet is packet 58

Overall Delay = Timestamp of Packet 58 − Timestamp of Packet 32

= 3.609706 - 3.606623

= 0.003083 = 0.0031 seconds

**Throughput of File 2. Txt:**

Total data transferred = 29,758 bytes

Overall delay = 0.0031 seconds

Throughput **=** Amount of data transferred / Overall delay

**= (**29758 \* 8)/0.0031

= 76794838.7097 bps

**Overall delay of File 3.txt:**

According to the Wireshark traces of File 3.txt (48 KB):

Initial SYN Packet is packet 102

Final ACK Packet is packet 146

Overall Delay = Timestamp of Packet 146 − Timestamp of Packet 102

= 0.021312-0.017271

= 0.003737 = 0.0037 seconds

**Throughput of File 3. Txt:**

Total data transferred = 44,642 bytes

Overall delay = 0.0037 seconds

Throughput **=** Amount of data transferred / Overall delay

**= (**44642 \* 8) /0.0037

= 96523243.2432 bps

**Overall delay of File 4.txt:**

According to the Wireshark traces of File 4.txt (64 KB):

Initial SYN Packet is packet 14

Final ACK Packet is packet 66

Overall Delay = Timestamp of Packet 66 − Timestamp of Packet 14

= 2.372713 - 2.365997

= 0.006716 = 0.0067 seconds

**Throughput of File 4. Txt:**

Total data transferred = 59,522 bytes

Overall delay = 0.0067 seconds

Throughput **=** Amount of data transferred / Overall delay

**= (**59522 \* 8) /0.0067

= 71071044.7761 bps

**Overall delay calculation using the Wireshark traces for UDP protocol:**

The process I followed for calculating the overall delay of data transfer over UDP protocol involves the following steps:

1. Open the Wireshark capture file: Begin by opening the Wireshark traces for the specific file you wish to analyze.

2. Apply a filter: Filter the traces using the port number on which the server is operating to narrow down the packets to the specific communication (udp.port == 20000)

3.Identify the initial packet: Identify the initial packet where the server initiates the data transfer.

4. Identify the final packet: Identify the final packet where the client acknowledges the data transferred by server.

5. Calculate Overall Delay: Determine the overall delay of data transfer by subtracting the timestamp of the initial packet from the timestamp of the final packet.

6. Calculate total data transferred: Sum up the payload lengths of all the data packets between initial packet and final packet, excluding the payloads of acknowledgement.

7. Calculate the throughput: Determine the throughput using the below formula

Throughput = Amount of data transferred/Total time taken

= Amount of data transferred/Overall delay

**Overall delay of File 1.txt:**

According to the Wireshark traces of File 1.txt (16 KB):

The initial Packet is packet 28

The final Packet is packet 57

Overall Delay = Timestamp of packet 28 − Timestamp of packet 57

= 6.008118 - 5.976440

= 0.031678 = 0.0317 seconds

**Throughput of File 1.txt:**

Total data transferred = 14878 bytes

Overall delay = 0.0317 seconds

Throughput **=** Amount of data transferred / Overall delay

**= (**14878 \* 8) /0.0317

= 3754700.31546 = 3754700.3155 bps

**Overall delay of File 2.txt**

According to the Wireshark traces of File 2.txt:

Initial SYN Packet is packet 62

Final ACK Packet is packet 121

Overall Delay = Timestamp of Packet 121 − Timestamp of Packet 62

= 2.428505 - 2.377667

= 0.050838 = 0.0508 seconds

**Throughput of File 2.txt:**

Total data transferred = 29,758 bytes

Overall delay = 0.0508 seconds

Throughput **=** Amount of data transferred / Overall delay

**= (**29758 \* 8)/0.0508

= 4686299.2126 bps

**Overall delay of File 3.txt**

According to the Wireshark traces of File 1.tcp:

Initial packet is packet 13

Final packet is packet 102

Overall Delay = Timestamp of Packet 102 − Timestamp of Packet 13

= 2.585560 - 2.505514

= 0.080046 = 0.0800 seconds

**Throughput of File 3.txt:**

Total data transferred = 44,642 bytes

Overall delay = 0.0800 seconds

Throughput **=** Amount of data transferred / Overall delay

**= (**44642 \* 8)/0.0800

= 4464200 bps

**Overall delay of File 4.txt**

According to the Wireshark traces of File 4.txt:

Initial packet is packet 43

Final packet is packet 162

Overall Delay = Timestamp of Packet 162 − Timestamp of Packet 43

= 3.391232 - 3.257172

= 0.13406 = 0.1341 seconds

**Throughput of File 4.txt:**

Total data transferred = 59,522 bytes

Overall delay = 0.1341 seconds

Throughput **=** Amount of data transferred / Overall delay

**=** (59522 \* 8)/0.1341

= 3550902.31171= 3550902.3117 bps

**Delay:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Delay | File 1 (16 KB) | File 2 (32 KB) | File 3 (48 KB) | File 4 (64 KB) |
| TCP (sec) | 0.0007 | 0.0031 | 0.0037 | 0.0067 |
| SNW (sec) | 0.0317 | 0.0508 | 0.0800 | 0.1341 |

**Throughput**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Throughput | File 1 (16 KB) | File 2 (32 KB) | File 3 (48 KB) | File 4 (64 KB) |
| TCP (sec) | 170034285.7140 | 76794838.7097 | 96523243.2432 | 71071044.7761 |
| SNW (sec) | 3754700.3155 | 4686299.2126 | 4464200 | 3550902.3117 |

**Trends observed:**

As file size increases, the TCP and SNW delay increases, which is due to the increased amount of data that needs to be transmitted.

The higher delay is observed in UDP when compared to TCP, this is due to the use of the Stop-and-Wait protocol in UDP. SNW mechanism involves sending one data chunk of 1000 bytes and waiting for acknowledgment before sending the next chunk, which can introduce additional delays. In contrast, TCP employs more advanced techniques selective acknowledgments, allowing it to send multiple packets before requiring acknowledgment.

The trends in throughput indicate that both TCP and SNW as the file size increases, the throughput decreases, which is likely due to the larger amount of data that needs to be transmitted. TCP achieves higher throughput than SNW because TCP uses windowing and flow control mechanisms, allowing it to send multiple packets before requiring an acknowledgment. This optimizes the use of available bandwidth and contributes to higher throughput.

File 2, File 4 experiences low throughput in TCP because of out of order delivery of packets. When packets arrive out of order, it may trigger the need for retransmissions, which can introduce delays in the data transfer process.

The increased delay for File 2 in UDP could be due to various factors, including network conditions and congestion.